

# **CREATIVE AND INNOVATIVE PROJECT (CIP) REPORT SCULPTURE MANAGEMENT SYSTEM**

**SUBMITTED BY**

VENKATA SURYA .V

11209A020

ARAVIND .S

11209M001

**GUIDED BY**

MR V BALU

ASSISTANT PROFESSOR



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SRI CHANDRASEKHARENDRASARASWATHI VISWA  
MAHAVIDYALAYA**

[MAY 2023]

# **Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya**

Enathur, Kanchipuram – 631501



## **BONAFIDE CERTIFICATE**

This is to certify that the CIP Report entitled **SCULPTURE MANAGEMENT SYSTEM** is the bonafide work carried out by Mr. VEMPARALA VENKATA SURYA, ARAVIND .S Reg.No 11209A020, 11209M001 during the academic year 2023-2024.

**MR V.BALU**  
Assistant Professor,  
Department of CSE,  
SCSVMV.

**DR .M. SENTHILKUMARN**  
Associate professor & Head  
Department of CSE,  
SCSVMV.

Submitted for the CIP project work viva - voce examination held on \_\_\_\_\_

Place : Kanchipuram.

Date :

*Examiner 1*

*Examiner 2*

## **DECLARATION**

It is certified that the CIP work titled **Sculpture Management System** is originally implemented by me. No ideas, processes, results or words of others have been presented as my own work. Due acknowledgement is given wherever others' work or ideas are utilized.

- a. There is no fabrication of data or results which have been compiled /analyzed.
- b. There is no falsification by manipulating data or processes, or changing or omitting data or results.

I understand that the project is liable to be rejected at any stage (even at a later date) if it is discovered that the project has been plagiarized, or significant code has been copied. I understand that if such malpractices are found, the project will be disqualified, and the Degree awarded itself will become invalid.

Signature of the student with date

## TABLE OF CONTENTS

Chapter No.	Title	Page No.
	<b>ABSTRACT</b>	<b>I</b>
	<b>LIST OF TABLES</b>	<b>II</b>
	<b>LIST OF FIGURES</b>	<b>III</b>
	<b>LIST OF ABBREVIATION</b>	<b>IV</b>
<b>1.</b>	<b>Introduction</b>	<b>1</b>
	1.1 Introduction	7
	1.2 Scope of the project	8
	1.3 Existing System	9
<b>2.</b>	<b>Literature Survey</b>	
	2.1 Literature Survey	10
	2.2 Problem Statement	11
<b>3.</b>	<b>Proposed Method / Architecture / Process /Methodology / Project Description</b>	
	3.1 Proposed method	12
	3.2 Architecture	13
	3.3 project Description	15
	3.4 Module Description	17
	3.5 Methodology	19
<b>4.</b>	<b>Implementation Work</b>	
	4.1 Implementation process	20
	4.2 Code	21
	4.3 Testing and methodologies	37

4.4	Result Analysis / Result comparison	38
5.	<b>Conclusion</b>	42
6.	<b>Future Enhancement</b>	43
7.	<b>References</b>	44

## Abstract

In this project, developed a system as a search engine for finding the temple sculpture details. Using this system we can search for an image of a sculpture, the system will provide the entire sculpture details like the name of the sculpture, temple name, state, and city of the temple. Once the image is searched into the database the comparison of the images is done with three parameters keyword, temple. First, the keyword of the image are compared, then if there's a match with all parameters only then the relevant image with data is extracted from the database, or else it is shown to the user the sculpture is not found in the database.

# CHAPTER 1

## Introduction

### 1. Introduction

A sculpture management search engine is a digital platform designed to assist art collectors, museum curators, and other individuals involved in the art world in the organization, tracking, and management of sculptures. Sculptures are unique and valuable works of art that require special attention and care, and a search engine specifically designed for managing sculptures can be an invaluable tool for those responsible for their upkeep.

**Functionality:** The search engine works by compiling information about various sculptures from temple . This information can include details about the materials used and current location. The search engine also includes features for managing sculptures within a collection. For example, users can track the conservation needs of each sculpture, as well as its exhibition history and provenance. The search engine may also include tools for managing loans and acquisitions, as well as for tracking the movement of sculptures between different locations.

**User Interface:** The user interface of the sculpture management search engine is designed to be intuitive and user-friendly. Users can easily search for sculptures by keyword , state or other relevant criteria. The search engine also includes features for browsing through sculptures by category or type, such as figurative sculptures, abstract sculptures, or sculptures made from a particular material.

**Security:** As sculptures are often unique and valuable works of art, security is a critical concern for any sculpture management search engine. The platform should include robust security measures to prevent unauthorized access and to protect against theft or damage.

The search engine may include features as managing access to sensitive information about each sculpture, such as its provenance or conservation needs.

**Data Analysis:** The sculpture management search engine can provide valuable insights and analytics about each sculpture in a collection. For example, the search engine may be able to identify patterns in exhibition history or provenance, allowing users to better understand the history and significance of each sculpture.

## 1.1 Scope of the project

The search engine for finding temple sculpture details would involve developing a software system that provides a user-friendly interface for searching temple sculptures. Here are some possible features and functionalities that could be included:

**Image Upload:** The system should provide an interface for temple administrators to upload images of present available sculptures. The system will compare keyword with the images available in the database using parameters like state, temple name.

**Database Management:** The system should have a database to store information about temple sculptures, such as the name of the sculpture, temple name, state, and city of the temple. The database should also include images of the sculptures and relevant information about them.

**Search Engine:** The search engine should be capable of comparing the parameters of the uploaded image for the temples available in the database. If the parameters match, then the relevant information about the sculpture should be displayed to the user.

**Reporting:** The system should generate reports on the search results, such as the number of records, records with respective state.



## **1.2 Existing system**

In existing there is no proper system for searching of temple sculptures details .If we want to search some temple sculpture in the web ,only the image is displayed to the user. It doesn't display any relevant details of the image for what we searched. So this system is very useful to police department to track the sculpture from which temple and description about the sculpture.

## **Chapter 2**

### **Literature survey**

#### **2.1 Literature survey**

**"Node.js and My SQL for Sculpture Management Portal" by Prasanth M. and Nithya N.**

In this article, the authors describe the development of a sculpture management portal using Node.js and My SQL as the backend database. They also use HTML, CSS, and Bootstrap for the frontend.

**"Using Node.js to Create Web Applications" by Mark Brown.**

This article provides a general introduction to Node.js and how it can be used to create web applications. Overall, there are many resources available that can help guide the development of a sculpture management portal using Node.js and HTML

**"Developing a Web-Based Sculpture Management System Using Node.js and React" by S. Y. Lau and C. K. Pang.**

This paper describes the development of a sculpture management system using Node.js and React, which allows users to manage sculptures, galleries, and artists.

**"A Web-Based Sculpture Management System Using Node.js and AngularJS" by Y. L. Wang and C. H. Lee.**

This paper describes the development of a sculpture management system using Node.js and AngularJS, which allows users to view sculptures and their related information.

**"Building a Web-Based Sculpture Management System Using Node.js and My SQL" by M. Z. Ahmed and M. F. Ahmed.**

This paper describes the development of a sculpture management system using Node.js and My SQL, which allows users to manage sculptures and their related information. The authors also discuss the system architecture and design, as well as the user interface.

## **2.2 Problem Statement**

Until now there is no existing search engine for sculpture management .The aim of the project to provide a search engine for the sculptures available in the temples and to provide the records for temples available in particular state. This new system should provide accurate and up-to-date information on sculptures. So this system is very useful to police department to track the sculpture from which temple and description about the sculpture.

## **Chapter 3**

### **3.1 Proposed method**

We propose developing a new and improved sculpture management console. The console will provide a centralized and automated system that is accessible to authorized personnel and provides accurate and up-to-date information on sculptures.

The console will be designed to be user-friendly and intuitive, with advanced search options and filters. Authorized personnel will be able to access and update this information quickly and efficiently, facilitating efficient management of sculptures.

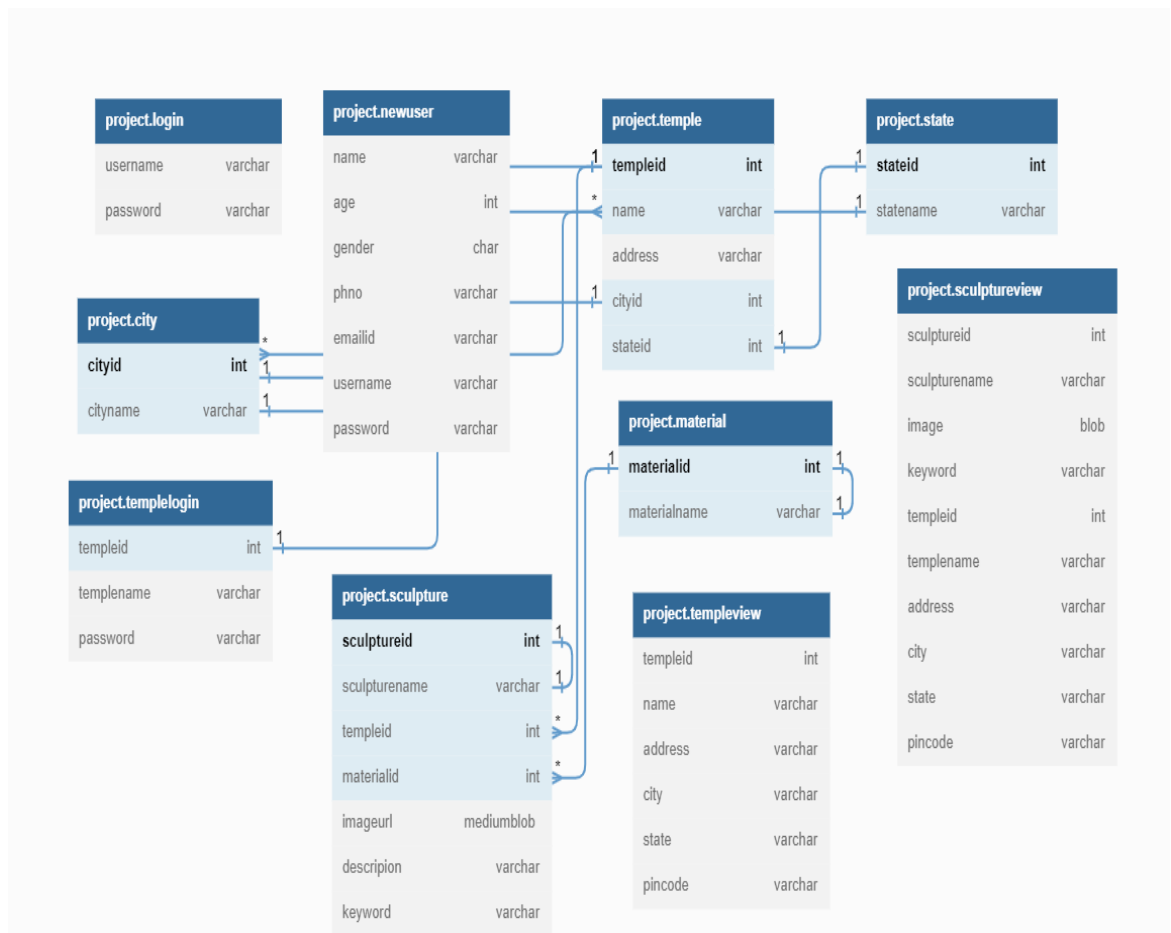
Overall, the proposed sculpture management console will provide an efficient and organized system for managing sculptures, with advanced features and functionality that will facilitate efficient and effective management of sculptures.

#### **Advantages of Proposed System**

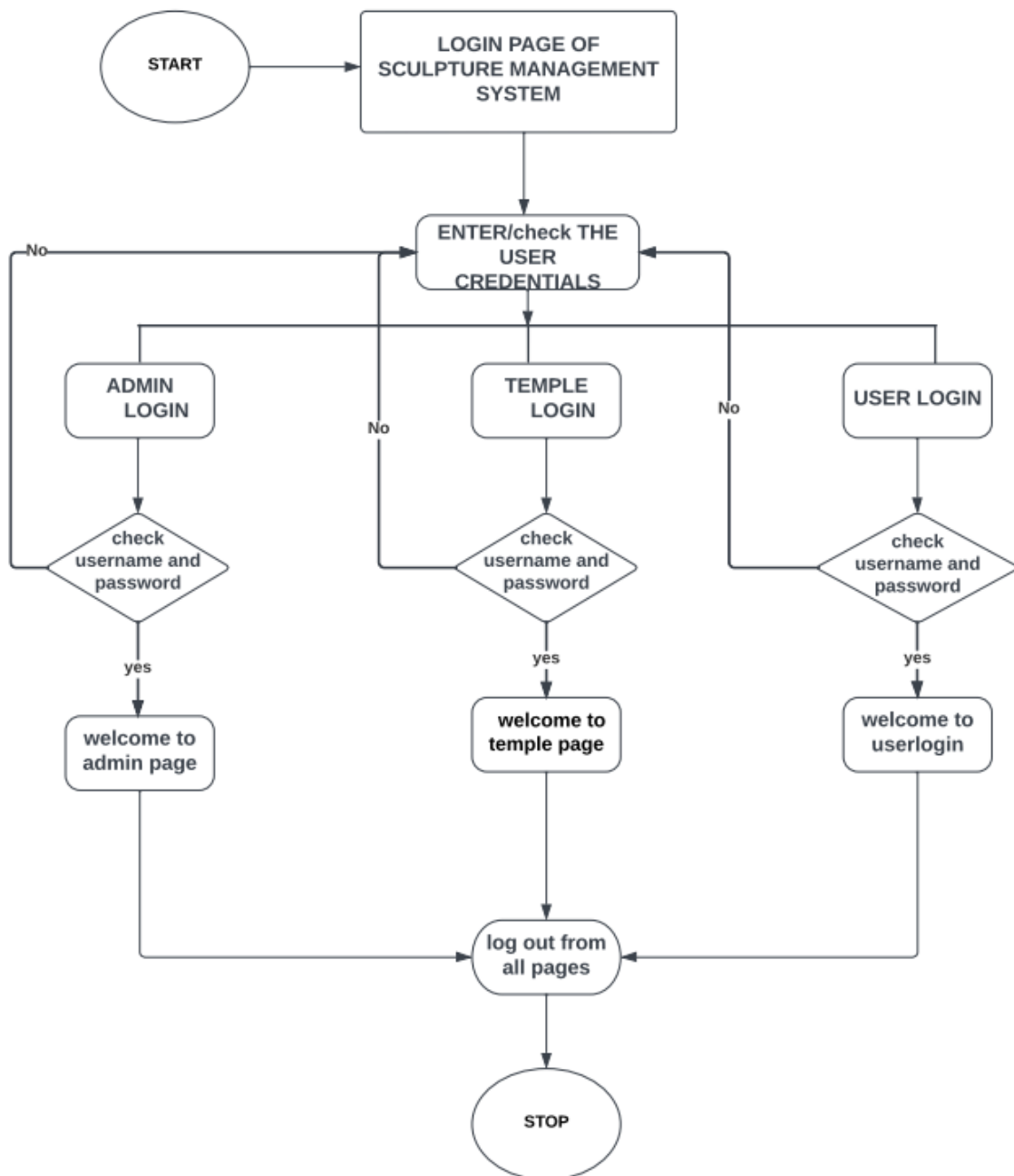
- Efficiency and Effectiveness
- It provide quick response with very accurate information
- Mainly used for the police department
- If any sculpture theft then retrieve the information about the sculpture is very easy

## 3.2 Architecture

### Database Design



## Architecture diagram



### **3.3 Project Description**

The Sculpture Management System is a web application designed to provide an efficient and organized system for managing sculptures. The system allows authorized personnel to access and update information related to sculptures.

The system provides a user-friendly interface for managing the details of sculptures and temples, including adding new temples, adding sculptures to existing temples, and updating and deleting existing temple and sculpture details. Users can also search for sculptures and temples by keyword, and view details about individual sculptures and temples.

If a sculpture has been stolen, the police department may need to gather information about the sculpture to aid in their investigation. This could include details such as the sculpture's ID, name, material, image, description, and keywords. Additionally, information about the location and time of the theft may also be important.

With a sculpture management console, the police department could easily access this information from a centralized database. They could search for a specific sculpture using its name, or generate reports on sculptures that match certain criteria, such as sculptures made from a specific material or sculptures located in a certain city.

Having this information readily available can help the police department quickly identify stolen sculptures, track them down, and return them to their rightful owners. It can also help prevent future thefts by enabling better tracking and monitoring of sculptures in public spaces.

The system stores details about temples, including the temple ID, name, address, city, state, and pincode. For each temple, the system can store multiple sculptures, with details including the sculpture ID, name, material used, an image of the sculpture, a description of the sculpture, and keywords to help with searching and filtering.

In addition, the system provides reporting functionality, allowing users to generate reports based on the information in the database. Reports can be generated based on various criteria, such as by temple or sculpture, or by city or state. Reports can be exported in various formats, such as PDF for further analysis and processing.



### 3.4 Module Description

The Sculpture Management System is an application designed to manage the details of sculptures in a collection or exhibition. This system contains three modules describe as follow

- Administrator module
- Temple module
- User module

**Administrator Module:** Administrator has a full authority to maintain the system with database. The Admin can login through adminlogin .The main functions of the administrator

- **Add temple :** Admin can add details about temple and assign one id for each temple . In this contain templeid ,name, address ,city ,state , pincode
- **Add sculpture :** Admin has accesses to add sculpture .It contain sculptureid,sculpturename,templeid,templename,image,description ,keyword.
- **Add TempleLogin :** Admin has only accesses to add Templelogin for particular temple .In this templeid, templename, username ,password.
- **Add Material :** Admin has to add any new material of the image required he has the authority to add material and assign one id to the material
- **Add City :** Admin has accesses to assign id for the city with the help of the id city can be accesable

**Temple Login:** In the temple Login he/she has the authority to access only temple page. In this module templelogin user can add sculpture related to that particular temple only

- **Add Sculpture :** Templeadmin has access to add sculpture . It contains sculptureid, sculpturename, templeid, templename, image, description, keyword. Here temple admin has authority to add multiple sculptures for a specific temple by linking them to the corresponding temple ID.

**User Login:** This module provides user authentication mechanism. User has to register with his details and then user gets the approval from admin then only the user will access the application. Main function of users is search .

- **Search :** In this search option user can search according to the keyword , state. This helps user to fetch the detail about temple sculpture easy for the police department in case of any theft happens.

### 3.5 Methodology

A sculpture management system is a software system designed to help manage and organize sculptures in a collection or exhibition. The methodology for developing a sculpture management system.

1. **Requirements gathering:** Identify the specific search criteria they need to find sculptures, such as the source of the sculpture (e.g., temple), the material used, state selection relevant to the collection. Determine how the search results displayed.
2. **Design:** Based on the requirements gathered, design the search engine module's architecture and user interface. The system should include modules for search query entry, filtering and sorting of results, and displaying detailed sculpture information.
3. **Implementation:** Develop the search engine module according to the design. This may involve selecting and configuring a search engine platform, creating the user interface, and developing the necessary backend components.
4. **Integration:** Integrate the search engine module with the sculpture management system to ensure that it can access the necessary data. Test the module to ensure that it meets the requirements and that the search results are accurate and relevant.
5. **Maintenance:** Provide ongoing maintenance and support for the search engine module. This may involve fixing bugs, optimizing search performance, and updating the module to meet changing requirements.

## Chapter 4

### Implementation work

#### 4.1 Implementation Process

- 1. Design:** We design the architecture and user interface for the sculpture management console. This may involve creating wireframes and mockups to visualize the user interface and flow of the console. Identify the necessary components and technologies needed to implement the console.
- 2. Database design:** We design and implement the database schema for the sculpture , including tables and relationships and views.
- 3. User interface development:** We develop the user interface for the sculpture management console using the appropriate front-end technologies such as HTML, CSS, and JavaScript. Create views and templates for different pages of the console.
- 4. Back-end development:** We develop the back-end functionality of the console using Node.js server-side technologies . This includes implementing the data entry, search and detailed display of sculpture information.
- 5. User authentication and authorization:** We implement a user authentication and authorization system to ensure that only authorized users can access the sculpture management console.

## 4.2 Sample Code

```
const mysql=require ("mysql");
const express=require("express");
const bodyParser =require("body-parser");
const encoder = bodyParser.urlencoded();
const app=express();
const ejs = require('ejs');
const cookieParser = require('cookie-parser');

app.use(bodyParser.urlencoded({ extended: true }));
app.use(cookieParser());

// Set the view engine to EJS
app.set('view engine', 'ejs');

app.use(express.static("public"))

const connection=mysql.createConnection({
  host:"localhost",
  user:"root",
  password:"manager",
  database:"project",
  port:"3306"

});

//connect to the database

connection.connect(function(error){
  if(error) throw error
```

```

        else console.log("connected to the database successfully");

    });

    // user login

    app.get("/",function(req,res){

        res.sendFile(__dirname + "/homewithlogindropdown.html");

    });

    app.get("/userlogin.html",function(req,res){

        res.sendFile(__dirname + "/userlogin.html")
    });

    app.get("/adminlogin.html",function(req,res){

        res.sendFile(__dirname + "/adminlogin.html")
    });

    app.get("/templelogin.html",function(req,res){

        res.sendFile(__dirname + "/templelogin.html")
    });

    // user login
    app.post("/userlogin.html",encoder,function(req,res){
        var username = req.body.username;
        var password = req.body.password;

        connection.query("select * from newlogin where username =? and password
        =?",[username,password],function(error,results,fields){

```

```

    console.log("select * from newlogin where username =" + username + "");

    if(results.length > 0){
        res.redirect("/home");
    }else{
        res.redirect("/homewithlogindropdown");
    }
    res.end();
})
})

// Set up the search route
// Set up the search route
app.get('/search', (request, response) => {
    const search = request.query.search;
    const search1 = request.query.search1;
    const searchBy = request.query.searchBy;
    console.log(searchBy);
    console.log(search);
    let sql = "";

    switch (searchBy) {
        case 'keyword':
            sql = `SELECT * FROM sculptureview WHERE keyword LIKE '%${search}%'`;
            break;
        case 'state':
            sql = `SELECT * FROM sculptureview WHERE keyword LIKE '%${search}%'
AND state LIKE '%${search1}%'`;
            break;
        case 'name':
            sql = `SELECT * FROM sculptureview WHERE name LIKE '%${search}%'`;
            break;
    }

```

```

    default:
        sql = `SELECT * FROM sculptureview WHERE keyword LIKE '%${search}%'`;
        break;
    }

    // execute query
    connection.query(sql, (err, result) => {
        if (err) {
            throw err;
        }
        //decode the image
        result.forEach((result)=> {
            result.image = Buffer.from(result.image, 'base64');
        });

        //render the search page
        response.render('search', { results: result });
    });
});

//when login is success
app.get("/home",function(req,res){

    res.sendFile(__dirname + "/userhome.html")
});

app.get("/homewithlogindropdown",function(req,res){

    res.sendFile(__dirname + "/homewithlogindropdown.html")
});

app.get("/imagegallery",function(req,res){

    res.sendFile(__dirname + "/imagegallery.html")
}

```



```

});
app.get("/contact-us.html",function(req,res){

    res.sendFile(__dirname + "/contact-us.html")
});
app.get("/imagepage.html",function(req,res){

    res.sendFile(__dirname + "/imagepage.html")
});

// regstration details
app.get("/register",function(req,res){

    res.sendFile(__dirname + "/Registration.html");
});

app.post("/register",encoder,function(req,res){
const { name,age,gender,phonenumber, emailid,username,password } = req.body;
const sql = 'INSERT INTO newlogin
(name,age,gender,phonenumber,emailid,username,password) VALUES (?,?, ?,
?,?,?,?)';

const values = [name,age,gender,phonenumber, emailid, username,password];

connection.query(sql, values, (err, result) => {
    if (err) {
        throw err;
    }
    console.log(`Message from ${name} (${emailid}) saved to database...`);
    res.redirect('/thankyou.html');
});
});

// when sucess

```

```

app.get("/thankyou.html",function(req,res){

    res.sendFile(__dirname + "/thankyou.html")
});

// admin login
app.post("/adminlogin.html",encoder,function(req,res){
    var username = req.body.username;
    var password = req.body.password;

    connection.query("select * from login where username =? and password
    =?",[username,password],function(error,results,fields){
        console.log("select * from login where username =" + username + "");

        if(results.length > 0){
            res.redirect("/adminhome");
        }else{
            res.redirect("/homewithlogindropdown");
        }
        res.end();
    })
})

//when login is success
app.get("/adminhome",function(req,res){

    res.sendFile(__dirname + "/adminhome.html")
});

app.get("/homewithlogindropdown",function(req,res){

    res.sendFile(__dirname + "/homewithlogindropdown.html")
});

```

```

app.get("/home.html",function(req,res){

    res.sendFile(__dirname + "/home.html")
});
app.get("/gettempleintemple.html",function(req,res){

    res.sendFile(__dirname + "/gettempleintemple.html")
});

app.get("/master.html",function(req,res){

    res.sendFile(__dirname + "/master.html")
});
// temple login
app.post("/templelogin.html",encoder,function(req,res){
    const templeid = req.body.templeid;
    const password = req.body.password;

    // execute MySQL query to validate user credentials
    const sql = `SELECT * FROM templelogin WHERE templeid = ${templeid} AND
password = '${password}'`;
    console.log("")
    connection.query(sql, (error, results, fields) => {
        if (error) {
            console.error(error);
            res.status(500).send('Error validating user credentials');
        } else {
            if (results.length > 0) {
                // user found with the given credentials
                const user = results[0];
                res.cookie('templeid', user.templeid);
                res.redirect('/templehome');
            } else {

```

```

        // no user found with the given credentials
        res.status(401).send('Invalid username or password');
    }
}
});
//when login is success
app.get("/templehome",function(req,res){

    res.sendFile(__dirname + "/templehome.html")
});

app.get("/homewithlogindropdown",function(req,res){

    res.sendFile(__dirname + "/homewithlogindropdown.html")
});
////////////////////////////////////
////////////////////////////////////index.html////////////////////////////////////
<!DOCTYPE html>
<html>
<head>
<title> Home Page</title>
<style>
    body {
margin: 0;
padding: 0;
font-family: Arial, sans-serif;
    }

    nav {
background-color: #333;
color: #fff;
display: flex;

```

```
    justify-content: space-between;
    align-items: center;
    padding: 10px;
}
```

```
.logo-container {
    display: flex;
    align-items: center;
}
```

```
.logo {
    height: 50px;
    margin-right: 10px;
}
```

```
.site-title {
    margin: 0;
    font-size: 24px;
}
```

```
.navigation {
    list-style-type: none;
    margin: 0;
    padding: 0;
    display: flex;
    align-items: center;
}
```

```
.navigation li {
    margin-right: 20px;
    border-radius: 30px;
    background: #69bada;
}
```

```
.navigation li a {  
    color: #fff;  
    text-decoration: none;  
    font-size: 18px;  
    display: block;  
    padding: 10px 20px;  
}  
  
.navigation li a: hover {  
    text-decoration: underline;  
}  
  
.dropdown {  
    position: relative;  
    margin-right :10px;  
}  
  
.dropbtn {  
    color: #fff;  
    font-size: 18px;  
    border: none;  
    background-color: inherit;  
    cursor: pointer;  
    margin-right :10px;  
}  
  
.dropdown-content {  
    display: none;  
    position: absolute;  
    background-color: black;  
    min-width: 140px;  
    z-index: 1;
```

```
}

.dropdown-content a {
  color: #333;
  padding: 12px 14px;
  text-decoration: none;
  display: block;
}

.dropdown-content a:hover {
  background-color: lightsalmon;
}

.dropdown:hover .dropdown-content {
  display: block;
  position: absolute;
  right: 0px;
}

.hero {
  background-color: #f1f1f1;
  text-align: center;
  padding: 100px 0;
  max-width: 100%;
  height: auto;
}

.bottomcenter {
  position: absolute;
  top: 100px;
  right: 300px;
}

.hero h1 {
  font-size: 38px;
```

```
margin: 0;
}

.hero p {
  font-size: 24px;
  margin-top: 20px;
}

footer {
  background-image: url('images/home2.jpg');
  background-repeat: no-repeat;
  background-size: 100%;
  background-color: #d65353;
  color: rgb(216, 53, 67);
  text-align: center;
  padding: 10px;
  flex: auto; /* decrease the padding value */
  position: relative;
  bottom: 0;
  left: 0;
  right: 0;

}

.footerimage {
  background-image: url('images/logo.jpg');
  background-repeat: no-repeat;
  background-size: 10%;
  display: block;
  margin-right: 10px;
  margin-bottom: 0px;
  position: relative;
  font-size: 20px;
```



```

    }

</style>
</head>
<body>
    <nav>
        <div class="logo-container">
            
            <h1 class="site-title">SCULPTURE MANAGEMENT SYSTEM
            </h1>
        </div>
        <ul class="navigation">
            <li><a href="home.html">Home</a></li>
            <li><a href="#">About Us</a></li>
            <li><a href="userlogin.html">Services</a></li>
            <li><a href="contact-us.html">Contact Us</a></li>
            <li class="dropdown">
                <a href="#" class="dropbtn">Login</a>
                <div class="dropdown-content">
                    <a href="userlogin.html">User Login</a>
                    <a href="adminlogin.html">Admin Login</a>
                    <a href="templelogin.html">Temple Login</a>
                </div>
            </li>
        </ul>
    </nav>
    <div class="hero">
        
        <div class="bottomcenter">
            <h1><b>SCULPTURE MANAGEMENT CONSOLE</b></h1></div>
            <br></br>
            <marquee direction="right" height="50" bgcolor= skyblue width="550"

```

scrollamount="100" scrolldelay="3000" behavior="alternate" loop="4"  
style="color:rgb(255, 255, 255)">

<h3><b>SCULPTURES ARE THE PRIDE OF OUR  
COUNTRY</b></h3>

</marquee>

</div>

<h3 align="center" color="purple";><b>IMPORTANCE OF TEMPLES IN  
INDIA</b></h3>

<p align="justify" style="text-indent:50px;background-color:red;color:white;line-height:25px;padding:10px;font-size:18px;letter-spacing:1px;border-radius:30px">A Hindu temple, or mandir or koil or kovil in Indian languages, is a house, seat and body of divinity for Hindus. It is a structure designed to bring human beings and gods together through worship, sacrifice, and devotion. The symbolism and structure of a Hindu temple are rooted in Vedic traditions, deploying circles and squares. It also represents recursion and the representation of the equivalence of the macrocosm and the microcosm by astronomical numbers, and by "specific alignments related to the geography of the place and the presumed linkages of the deity and the patron". A temple incorporates all elements of the Hindu cosmos — presenting the good, the evil and the human, as well as the elements of the Hindu sense of cyclic time and the essence of life — symbolically presenting dharma, artha, kama, moksha, and karma.</p>

<p align="justify" style="text-indent:50px;background-color:rgba(87, 136, 113, 0.329);color:rgb(29, 27, 27);line-height:25px;padding:10px;font-size:18px;letter-spacing:1px;border-radius:30px"><b>The spiritual principles symbolically represented in Hindu temples are given in the ancient Sanskrit texts of India (for example, the Vedas and Upanishads), while their structural rules are described in various ancient Sanskrit treatises on architecture (Bṛhat Saṃhitā, Vāstu Śāstras). The layout, the motifs, the plan and the building process recite ancient rituals, geometric



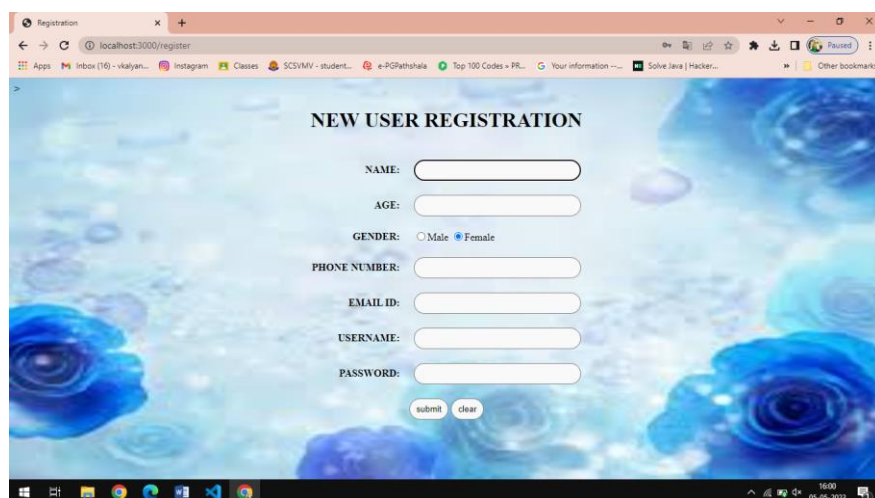
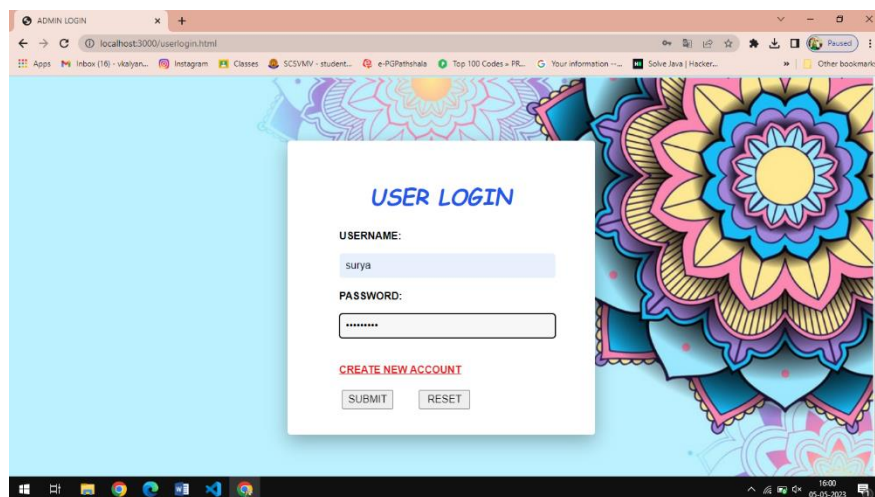
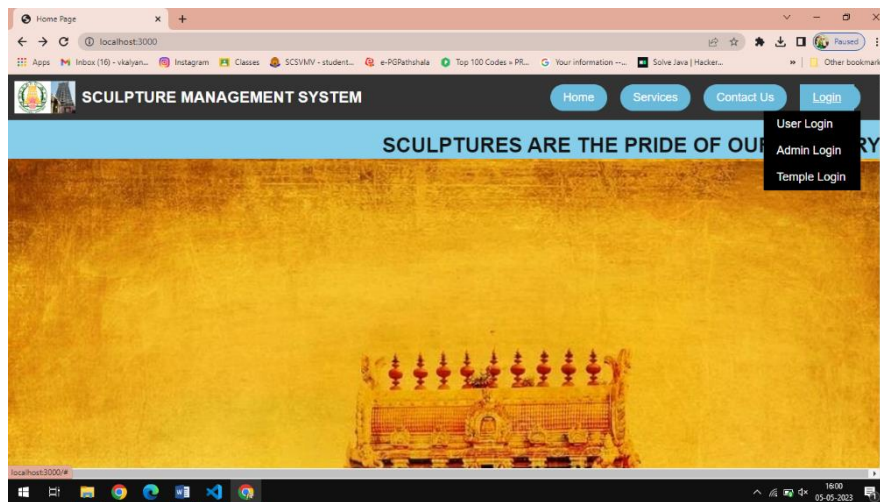
&lt;/html&gt;

### **4.3 Testing And Methodology**

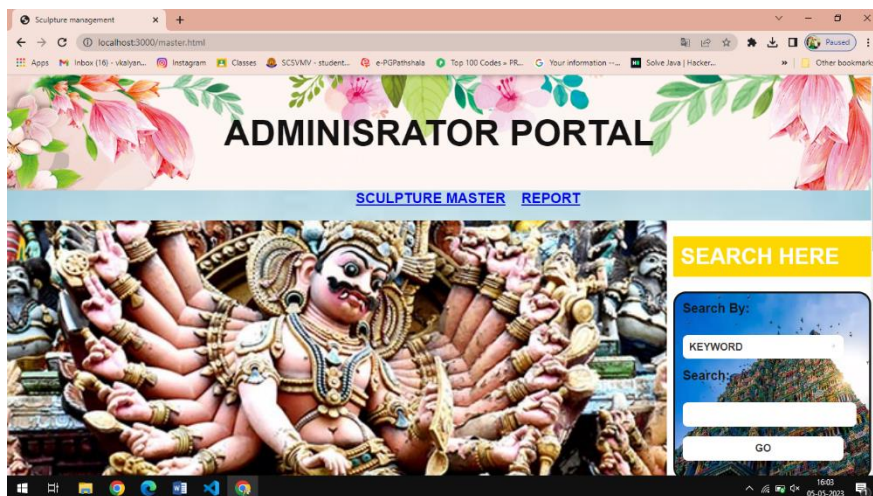
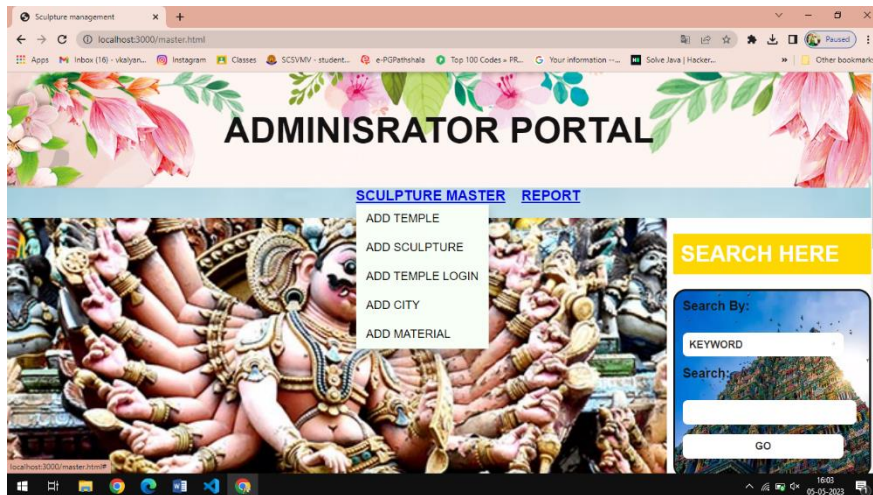
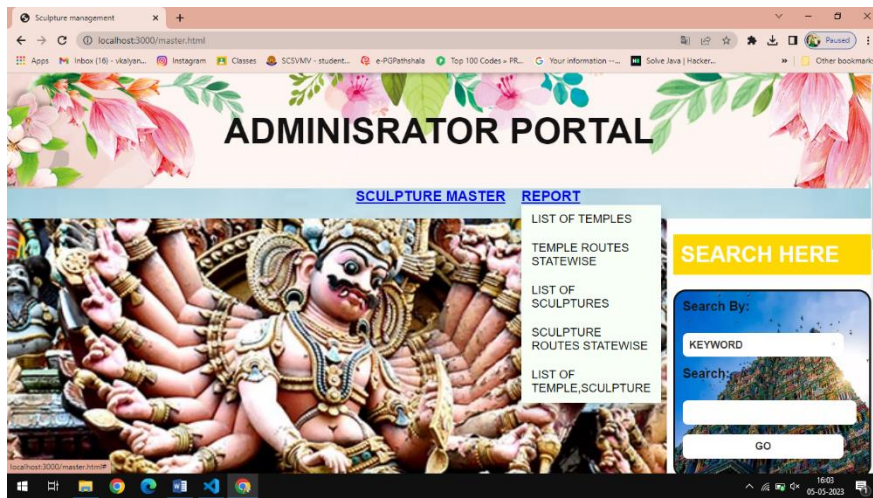
Testing is an important part of the development process for a sculpture management system, as it ensures that the system is working correctly and meets the requirements of its users.

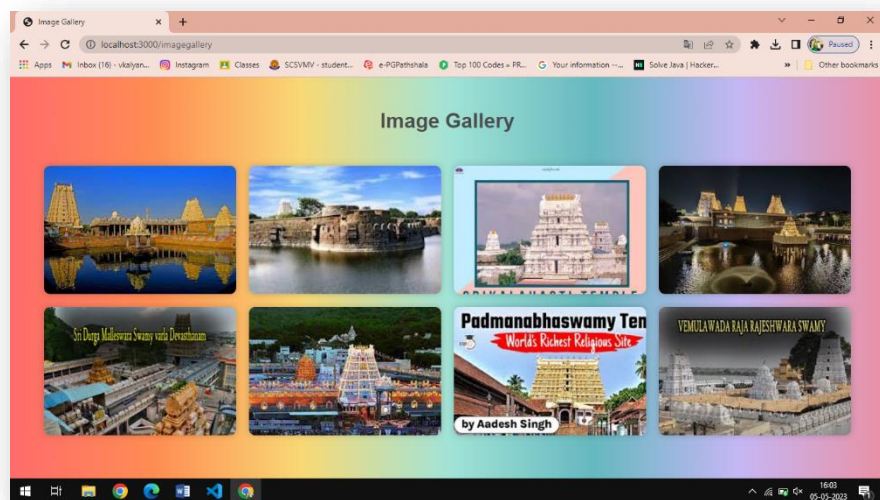
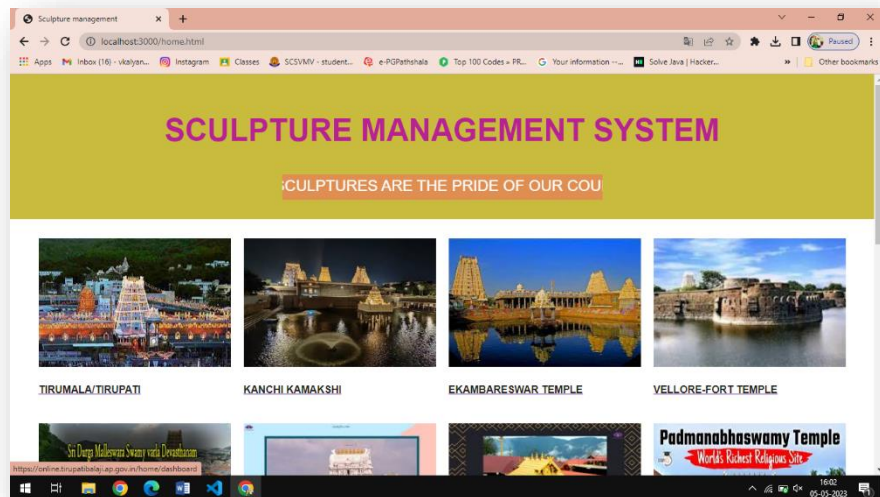
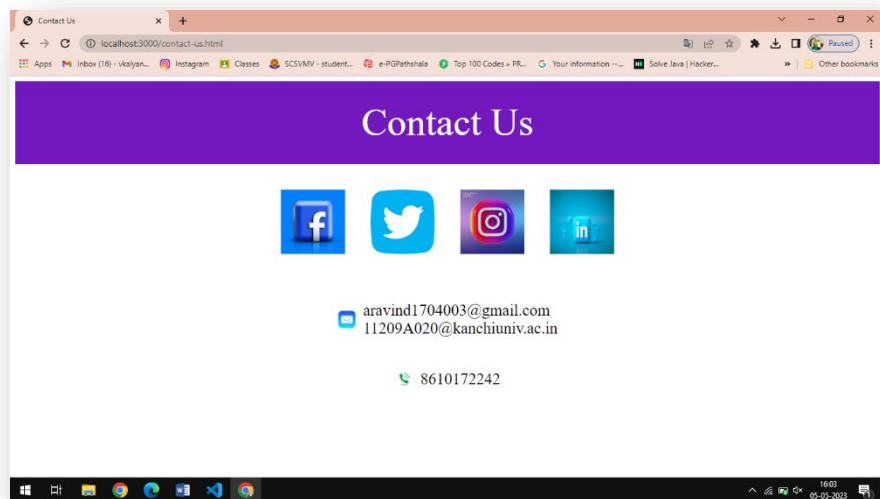
1. Unit testing: First we start test separate files to ensure that they work as expected.
2. Integration testing: Later we start test different components of the system work together as a whole. In this we include testing how the user interface interacts with the database
3. Overall testing : finally we deploy total project in localhost server and we solve all the vulnerability and bugs

## 4.4 Result

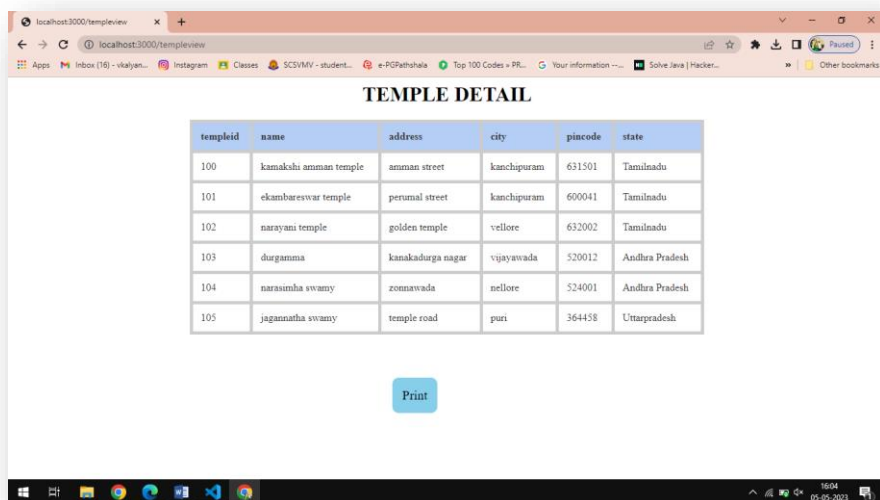
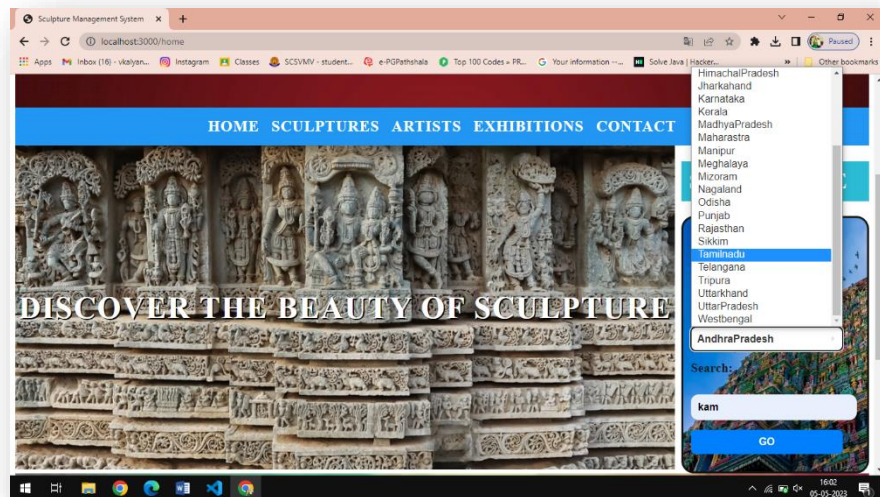
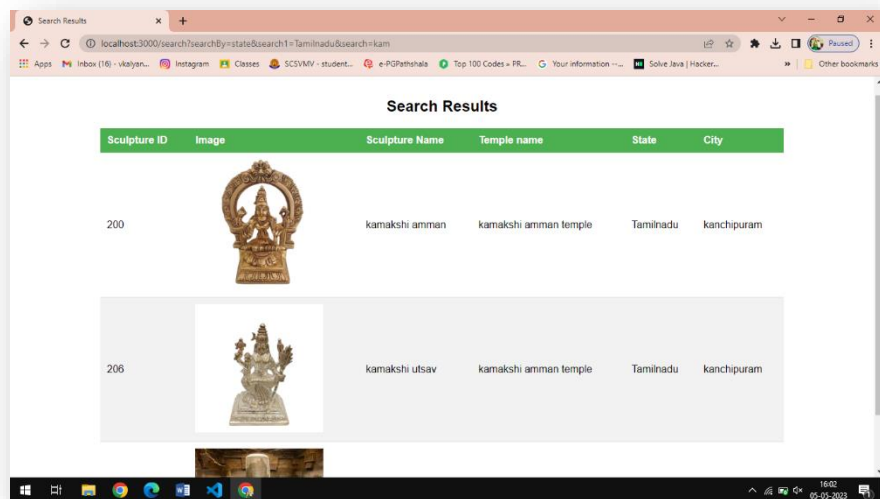












## Chapter 5

### 1.1 Conclusion

A search engine for sculpture management can be a valuable tool for the **police department** and other temple authority in the field of sculpture. Such a search engine would allow users to easily find and compare information about different sculptures, including their history, provenance, and current location. Additionally, it could help to streamline the process of managing and tracking sculptures within a collection, making it easier to keep track of important details such as conservation needs and exhibition history. Overall, a sculpture management search engine has the potential to greatly enhance the accessibility and organization of information related to sculptures. However, if the sculpture is identified as belonging to a temple, it is likely that further research could be done to trace its history and cultural significance.

## **CHAPTER 6**

### **6.1 FUTURE ENHANCEMENT**

In the future, we have plan to enhance the sculpture management system with advanced features and functionality. One of the key areas of improvement is in the field of image comparision. Based on the size, pixel, color the image is compared. We aim to implement sharp image comparison techniques to provide a more comprehensive and accurate analysis of sculptures. The uploaded image is compared with the database existing image if the match found then it display the details of the particular uploaded sculpture .This will help the police department to fetch the image details easily from which temple and state.

## Chapter 7

### 7.1 References

- [1] Rao, T. A. G., Elements of Hindu Iconography, Vol. I, Part. I, The Law Printing House, Madras: 1914, p. 37
- [2] Sastri, H. K., South Indian Images of Gods and Goddesses, Government of Madras, Madras: 1916, p. 89
- [3] Khandalavala, K.J., Indian Sculpture and Painting: An Introductory Study, D.B. Taraporevala Sons & Company, Bombay: 1983, p. 11
- [4]. Krishna Kumar Pandey<sup>1</sup> , Nishchol Mishra<sup>2</sup> ,Hemant kumar sharma  
Enhanced of color matching algorithm for image retrieval IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011.
- [5]. V.S.V.S. Murthy, E.Vamsidhar, J.N.V.R. Swarup Kumar “Content Based Image Retrieval using Hierarchical and K-Means Clustering Techniques “ 2010.
- [6]. Sangoh Jeong, “Histogram-Based Color Image Retrieval”, 2001

